Detecting suspicious new instances in your AWS Elastic Compute Cloud (EC2) environment

Applicability

- Product: Splunk Cloud Platform or Splunk Enterprise
- Feature: Monitoring
- Technology: Splunk App for AWS and Splunk Add-on for AWS
- Data: Amazon Web Services: CloudTrail and CloudWatch

Scenario

You are an Amazon Web Services (AWS) admin who manages AWS resources and services across your organization. You are aware that cryptomining or cryptojacking on your environment is a potential problem you need to be aware of. When malicious miners appropriate a cloud instance, often spinning up hundreds of new instances, the costs can become heavy for the account holder. You need to monitor your systems for suspicious activities that could indicate that your network has been infiltrated.

These searches are designed to detect suspicious new instances in your Elastic Compute Cloud (EC2) environment to help prevent cryptomining or cryptojacking from occurring.

- Some commands, parameters, and field names in the searches below may need to be adjusted to match your environment. In addition, to optimize the searches shown below, you should specify an index and a time range when appropriate.
- To run these searches, install the AWS App for Splunk (version 5.1.0 or later) and Splunk Add-on for AWS (version 4.4.0 or later), then configure your CloudTrail inputs.

Support searches

► Previously seen AWS regions

This search looks for CloudTrail events where an AWS instance is started and creates a baseline of most recent time (latest) and the first time (earliest) you see this region in your dataset, grouped by the value awsRegion.

```
sourcetype=aws:cloudtrail StartInstances | stats earliest(_time) AS earliest latest(_time) AS latest BY awsRegion | outputlookup previously_seen_aws_regions.csv | stats count
```

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
Previously seen EC2 AMIs

This search builds a table of previously seen Amazon Machine Images (AMIs) used to launch EC2 instances.

```
sourcetype=aws:cloudtrail eventName=RunInstances errorCode=success
| rename requestParameters.instancesSet.items{}.imageId AS amiID
| stats earliest(_time) AS earliest latest(_time) AS latest BY amiID
| outputlookup previously_seen_ec2_amis.csv
| stats count
```

Previously seen EC2 instance types

This search builds a table of previously seen EC2 instance types.

```
sourcetype=aws:cloudtrail eventName=RunInstances errorCode=success
| rename requestParameters.instanceType AS instanceType
| fillnull value="m1.small" instanceType
| stats earliest(_time) AS earliest latest(_time) AS latest BY instanceType
| outputlookup previously_seen_ec2_instance_types.csv
| stats count
```

Previously seen EC2 launches by user

This search builds a table of previously seen Amazon Resource Names (ARNs) that have launched a EC2 instance.

```
sourcetype=aws:cloudtrail eventName=RunInstances errorCode=success
| rename userIdentity.arn AS arn
| stats earliest(_time) AS firstTime latest(_time) AS lastTime BY arn
| outputlookup previously_seen_ec2_launches_by_user.csv
| stats count
```

Contextual searches

Get EC2 launch details

To run this search, you'll need to configure your AWS description inputs.

This search returns some of the launch details for a EC2 instance.

```
s| search sourcetype=aws:cloudtrail responseElements.instancesSet.items{}.instanceId={dest}
s| rename userIdentity.arn AS arn, responseElements.instancesSet.items{}.instanceId AS instanceld, responseElements.instancesSet.items{}.privateIp AS privatelpAddress, responseElements.instancesSet.items{}.imageId AS amiID, responseElements.instancesSet.items{}.architecture AS architecture, responseElements.instancesSet.items{}.keyName AS keyName
| table arn, awsRegion, instanceld, architecture, privatelpAddress, amiID, keyName
```

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
Detection searches

► Abnormally high AWS instances launched by user

This search looks for CloudTrail events where a user successfully launches an abnormally high number of instances. In this search, you query CloudTrail logs for events where an instance is successfully launched by a particular user. Because you want to detect a high number of instances launched within a short period, you create event buckets for 10-minute windows. You then calculate the total number of instances launched by a particular user, as well as the average and standard deviation values. Assign a threshold value in the search. You can start with 3 but will likely need to adjust it for your environment. The `eval` function sets the outlier 1 if the number of instances is greater than the average number of instances terminated, added to the multiplied value of threshold and standard deviation. Keep only the outliers and then calculate the number of standard deviations the value is away from the average.

This search can result in false positives since many service accounts configured within an AWS infrastructure are known to exhibit this behavior. You'll need to adjust the threshold values and filter out service accounts from the output. Always verify if this search alerted on a human user.

```
sourcetype=aws:cloudtrail eventName=RunInstances errorCode=success | bucket span=10m _time | stats count AS instances_launched BY _time userName | eventstats avg(instances_launched) AS total_launched_avg, stdev(instances_launched) AS total_launched_stdev | eval threshold_value = 4 | eval isOutlier=if(instances_launched > total_launched_avg+(total_launched_stdev * threshold_value), 1, 0) | search isOutlier=1 AND _time >= relative_time(now(), "-10m@m") | eval num_standard_deviations_away = round(abs(instances_launched - total_launched_avg) / total_launched_stdev, 2) | table _time, userName, instances_launched, num_standard_deviations_away, total_launched_avg, total_launched_stdev
```

► EC2 instance started in previously unseen region

Run the [Previously seen AWS regions](#) support search only once to create a baseline of previously seen regions.

This search looks for CloudTrail events where an instance is started in a particular region in the last one hour and then compares it to a lookup file of previously seen regions where an instance was started.

In this search, you query CloudTrail logs to look for events that indicate that an instance was started in a particular region. Using the `previously_seen_aws_regions.csv` lookup file created using the support search, you compare the region where this instance was started to all previously observed regions. The `eval` and `if` functions determine that the earliest times seen for this region and instance were within the last day. If a new region is detected, it alerts you with [Instance Started in a New Region](#). This region will be added to the list of `previously_seen_aws_regions.csv`. You'll need to maintain `previously_seen_aws_regions.csv`.

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
This search works best when you run the [Previously seen EC2 AMIs](#) support search once to create a history of previously seen AMIs.

This search looks for EC2 instances being created with previously unseen AMIs. The subsearch returns the AMI ID of all successful EC2 instance launches within the last hour and then appends the historical data from the lookup file to those results. It then recalculates the earliest and latest seen time field for each AMI ID and returns only those AMI IDs that have first been seen in the past hour. This is combined with the main search to return the time, user, and instance id of those systems.

After a new AMI is created, the first systems created with that AMI will cause this alert to fire. Verify that the AMI being used was created by a legitimate user.

This search works best when you run the [Previously seen EC2 instance types](#) support search once to create a history of previously seen instance types.

This search looks for EC2 instances being created with previously unseen instance types. The subsearch returns the instance types of all successful EC2 instance launches within the last hour and then appends the historical data in the lookup file to those results. It then recalculates the earliest seen time field for each instance type and returns only

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
those instance types that have first been seen in the past hour. This is combined with the main search to return the
time, user, and instance id of those systems.

It is possible to get false positives on this search since an admin can create a new system using a new instance type
never used before. Verify that the creator intended to create the system with the new instance type.

```
sourceType=aws:cloudtrail eventName=RunInstances [search sourceType=aws:cloudtrail
eventName=RunInstances errorCode=success
| fillnull value="m1.small" requestParameters.instanceType
| stats earliest(_time) AS earliest latest(_time) AS latest BY requestParameters.instanceType
| rename requestParameters.instanceType AS instanceType
| inputlookup append=t previously_seen_ec2_instance_types.csv
| stats min(earliest) AS earliest max(latest) AS latest by instanceType
| outputlookup previously_seen_ec2_instance_types.csv
| eval newType=if(earliest >= relative_time(now(), "-65m@m"), 1, 0)
| convert ctime(earliest) ctime(latest)
| where newType=1
| rename instanceType AS requestParameters.instanceType
| table requestParameters.instanceType]
| spath output=user userIdentity.arn
| rename requestParameters.instanceType AS instanceType, responseElements.instancesSet.items{}.instanceId AS dest
| table _time, user, dest, instanceType
```

**► EC2 instance started with previously unseen user**

This search looks for EC2 instances created by users who have not created them before. The subsearch returns the
ARNs of all successful EC2 instance launches within the last hour and then adds the historical data in the lookup file
to those results. It then recalculates the firstTime and lastTime field for each ARN and returns only those
ARNs that have first been seen in the past hour. This is combined with the main search to return the time, user, and
instance id of those systems.

```
search sourceType=aws:cloudtrail eventName=RunInstances [search sourceType=aws:cloudtrail
eventName=RunInstances errorCode=success
| stats earliest(_time) AS firstTime latest(_time) AS lastTime by userIdentity.arn
| rename userIdentity.arn AS arn
| inputlookup append=t previously_seen_ec2_launches_by_user.csv
| stats min(firstTime) AS firstTime, max(lastTime) AS lastTime by arn
| outputlookup previously_seen_ec2_launches_by_user.csv
| eval newUser=if(firstTime >= relative_time(now(), "-1h@h"), 1, 0)
| convert timeformat="%m/%d/%Y %H:%M:%S" ctime(firstTime)
| convert timeformat="%m/%d/%Y %H:%M:%S" ctime(lastTime)
| rename arn AS userIdentity.arn
| table userIdentity.arn
| rename requestParameters.instanceType AS instanceType, responseElements.instancesSet.items{}.instanceld AS dest
| table _time, user, dest, instanceType
```

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and
all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the
completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
Additional resources

This use case is included within Splunk Enterprise Security, a Splunk app that provides prebuilt content and searches to help answer root-cause questions in real-time about malicious and anomalous events in your IT infrastructure. In addition, Splunk Enterprise Security provides a number of other searches to help reinforce your Cloud Security posture, including:

- Detecting AWS cross-account activity
- Detecting AWS network ACL activity
- Detecting AWS suspicious provisioning activities
- Monitoring user activity spikes in AWS
- Monitoring AWS EC2 for suspicious login activities
- Monitoring AWS S3 for suspicious activities
- Monitoring AWS for suspicious traffic
- Monitoring AWS EC2 for unusual modifications